References

[Batcher] K. E. Batcher, "Array Unit," in The Massively Parallel Processor , J. L. Potter, Ed., The MIT Press, 1985.

[Burkley] J. T. Burkley, "MPP VLSI Multiprocessor Integrated Circuit Design," in The Massively Parallel Processor , J. L. Potter, Ed., The MIT Press, 1985.

[GAC] Goodyear Aerospace Corporation, "Functional Description of the MPP PE," GER 16624, Revision 3, February 1983.

clear the shift register.

One performance option is still open and will not be resolved until VLSI design is underway. It is anticipated that on-chip propagation delays will be very small and the clock speed can be very high. Perhaps the difference between on-chip processing rates and the rate at which signals can cross chip boundaries can be a factor of two or four. The option is to design the chip with dual operation rates based on activity internal to the chip versus activity with signals crossing chip boundaries. Instructions are sent from a control unit and must cross chip boundaries. The idea is to use one instruction for multiple internal processing cycles. This would improve performance when instructions are repeated for multiple bit fields, as with arithmetic on multiple bit data items. This option is still under study.

6. Acknowledgements and Conclusion

The work reported in this paper resulted from the efforts of a group of researchers participating in this project at MCNC. In addition to the authors, these people included Fred Heaton (MCNC) and Jothy Rosenberg (Duke). We also benefitted from discussions with Kenneth Batcher of Loral Systems and Charles Fiduccia of General Electric.

The specifications of a processing element and a chip architecture for a massively parallel processor have been given. This report can be used as a requirements document for the BLITZEN PE VLSI design effort.

5. Architecture and Performance Options

During the development of the BLITZEN PE architecture various alternatives were considered. This section comments on some of the options.

One goal of this effort is to develop a system which is miniaturized as compared to other commercially available systems. To achieve a reduction in system size, it is desirable to eliminate the MPP staging memory. The staging memory buffers and reformats data. It does the corner-turning necessary for bit-serial processing. The BLITZEN system must support corner turning . One method considered was to have two S Planes running in orthogonal directions. This would allow movement of data in both East-West and North-South directions overlapped with other processing. In the final design an I/O bus was selected for the East-West movement. A modified North-South S plane, in which the S paths are limited to on-chip PE's, was considered as support for corner-turning. It was decided that off chip devices were better able to provide the function without occupying chip area.

A related consideration concerned the need for a routing network at the P register if one or two S planes were included. For performance, both routing and S planes are useful if data must be corner-turned on-chip. Data movement associated with I/O could be overlapped with processing. We intend to provide off-chip support for corner-turning.

Local, indirect, addressing is very desirable. Consideration was given to having an adder at each PE to modify the global address. This was rejected as too expensive in chip area. It would take a multiple bit adder just for address arithmetic when the processing element has only a single bit adder for execution. The decision was made to provide modification by a logical OR rather than add. Since any form of local addressing means local decoding of the address at each PE, it is still expensive in area and subject to revision as VLSI design of the chip proceeds.

The shift register was the subject of several discussions. In the MPP it is variable in length in steps of four bits. A processor is bit serial and can handle data of any length. For this reason consideration was given to making the shift register adjustable in steps of one bit. The notion was rejected since it would significantly increase chip area taken for the shift register and would add to the set of operations resulting in an increase in the number of operation code bits. An operation was added to

4.3 Package Pins

A summary of logic pin requirements for the BLITZEN chip is given in table 4-1. The operation code uses 23 pins as defined in section 3.2. The "valid" function is a chip enable control signal.

Function	Pins	Direction
Operation code	23	in
Memory address	10	in (two pins also used for column select)
Column select	2	in
Valid	1	in
I/O bus	32	in/out
Neighbor interconnects	48	in/out
Sum-OR	1	out

Table 4-1. Logic pin requirements.

A total of 117 pins is needed for logic signals to the 8 by 16 array of PE's on a chip. This does not include power or clock signals.

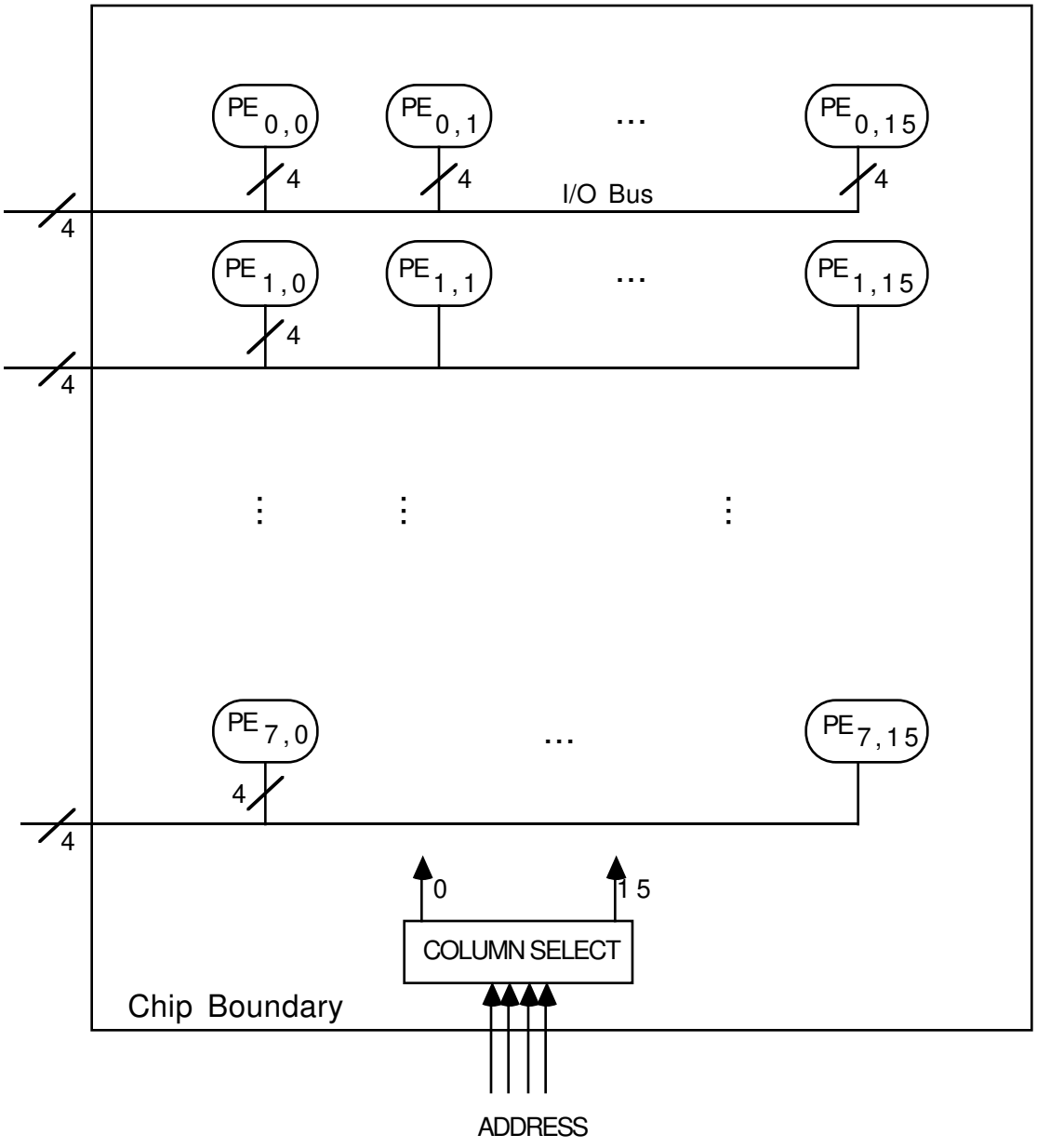

Figure 4-2. I/O bus interconnections.

a single bit wide path between each PE and its eight nearest neighbors. All PE's route the same direction in one processing cycle, according to the P register routing operation code. Eight paths can be established with four wires out of each PE by sending data on one wire, receiving data on one of the other three wires, and placing the remaining two wires in the high impedance state.

This X-shaped grid interconnects PE's on a chip and extends across chip boundaries so that an array of chips can be uniformly interconnected. Additional off-chip logic can provide various treatments of edges of the total array, as was done in the MPP system.

At the chip level, 48 pads are required for neighbor interconnects.

4.2 Input /Output Interconnections

The four bit wide I/O bus provides high bandwidth movement of data between on-chip memories and off-chip storage devices or other peripherals as shown in figure 4.2.

When performing I/O, all eight busses are active in the same direction. For I/O to particular PE's, it is necessary to select a column of PE's from the 16 columns on the chip by using a four bit address. Data may also be broadcast to all PE's in the row of a data bus. All columns are considered to be selected when broadcasting. Memories are addressed on four-bit boundaries by using the eight most significant bits of the ten bit memory address. I/O and memory read or write operations are not overlapped. This allows the use of the two least significant bits of the ten bit memory address to share the same chip pads as the two least significant bits of the column select address.

The I/O bus can be used for movement of four-bit data items across PE's on a chip in two operations. An output operation on one column can be followed by an input operation on a different column.

At the chip level, 32 pads are needed for I/O busses and four pads (two shared with memory address) are needed for column selection.

4. Custom Chip Architecture

Previous sections have defined the structure and set of operations for a single processing element (PE). Architecture of the custom VLSI integrated circuit for BLITZEN has 128 PE's organized logically into an array of 8 x 16 elements. This section defines the chip architecture.

4.1 Nearest Neighbor Interconnections

Processing elements are interconnected in various ways on a chip. Figure 4-1 shows the nearest neighbor routing network. This network provides

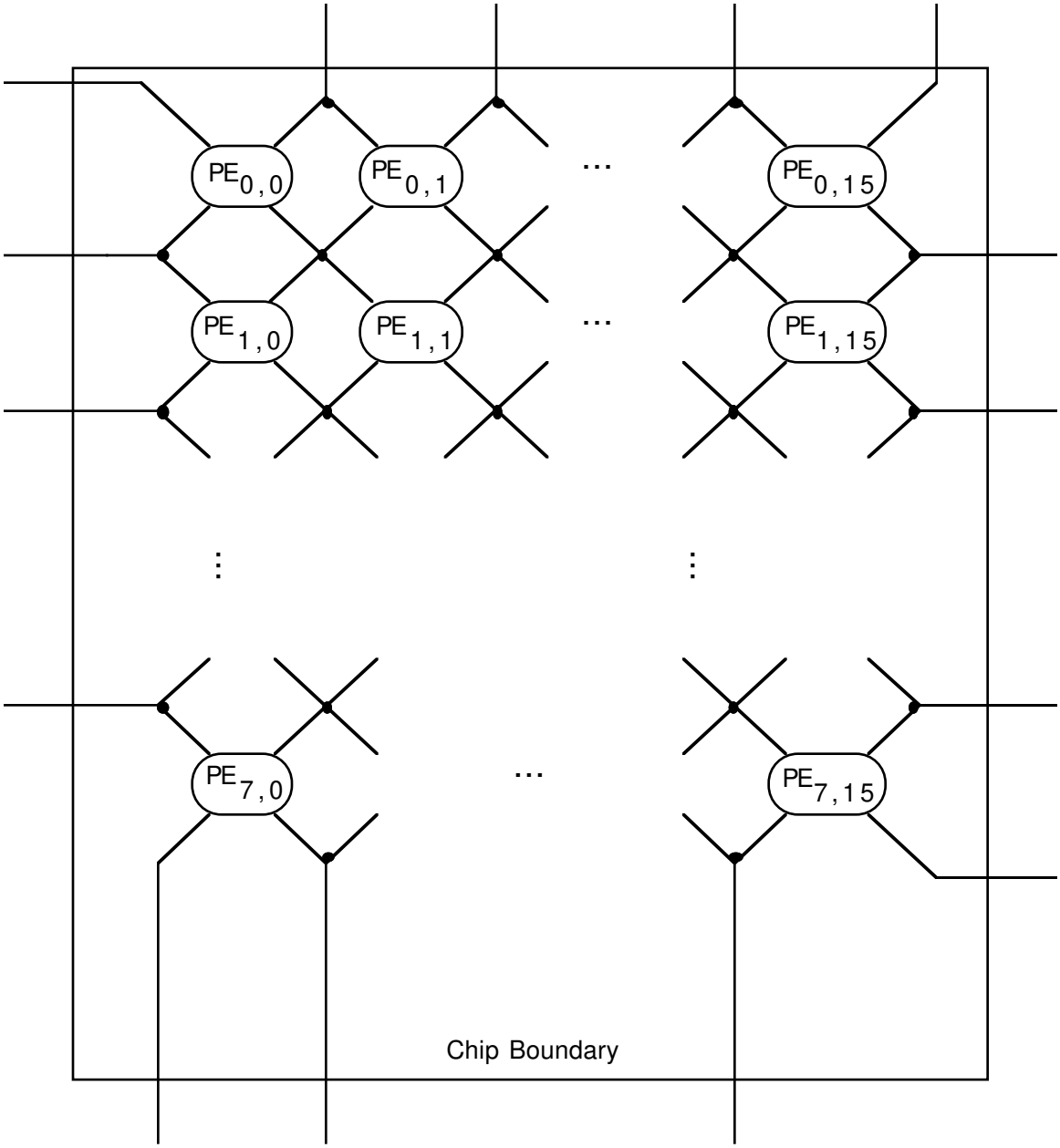

Figure 4-1. Nearest neighbor interconnections.

Category	Operation		Operation Code Control Vector							
			0 1 2	3 4 5 6 7	8 9 A	B C	D E	F G	H I	L M N O
Adder	(C,B) <-- P + A + C (C,B) <-- A + C Clear C Set C				1 1 0 1 0 0 1 0 1 1 1 1					- * - - - * - - - * # - - * # -
Shift register	Shift right Shift left Clear shift register					1 1 1 0 0 1				- * - - - * - - - * - -
	Set length N to:	2 6 10 14 18 22 26 30		1 1 0 0 0 1 1 0 0 1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1						
A Register	A <-- D A <-- SR, shift rt A <-- 0						1 0 1 1 0 1			- * - - - * - - - * - -
B Register	B <-- D B <-- SR, shift left B <-- 0				0 1 0 0 1 1 0 0 1					- * - - - * - - - * - -
G Register	G <-- D							0 1		
K Register	K <-- D							1 0		- * - -
Memory and I/O	M <-- D M <-- I/O bus M <-- I/O bus (broadcast) I/O bus <-- M								1 1 0 1 0 1 1 0	- * - @ - * 0 @ - * 1 @ - - - @
Sum -OR	Output sum-OR of D							1 1		

Table 3-1, continued. BLITZEN PE Operation Codes

cycle are: (1) read memory and place the bit on the data bus, (2) load P with the value from the data bus, (3) perform a full add, (4) shift the shift register right, and (5) transfer the shift register into A. The operation code for this, with commas separating fields, is: "111,01001,110,11,10,00,00,0000".

Category	Operation	Operation Code Control Vector							
		012	34567	89A	BC	DE	FG	HI	LMNO
Data bus source	A	001							
	B	010							
	C	011							
	G	100							
	K	101							
	P	110							
	M	111							---@
P register logic (D= data bus)	P <-- $\overline{D} \bullet P$		01111						*-#-
	P <-- $\overline{D} \bullet P$		01101						*-#-
	P <-- 0		00101						*-#-
	P <-- $\overline{D} \vee P$		01010						*-#-
	P <-- $\overline{D} \oplus P$		00010						*-#-
	P <-- $\overline{D}$		00111						*-#-
	P <-- $\overline{D} \bullet \overline{P}$		01100						*-#-
	P <-- $D \vee P$		01000						*-#-
	P <-- D		00011						*-#-
	P <-- $D \oplus P$		00110						*-#-
	P <-- $D \bullet \overline{P}$		01110						*-#-
	P <-- 1		00001						*-#-
	P <-- $D \vee \overline{P}$		01001						*-#-
	P <-- $\overline{D} \vee P$		01011						*-#-
	P <-- $\overline{P}$		00100						*-#-
P register routing	Route to:								
	N, P <-- P(S)		10000						*---
	NE, P <-- P(SW)		10001						*---
	E, P <-- P(W)		10010						*---
	SE, P <-- P(NW)		10011						*---
	S, P <-- P(N)		10100						*---
	SW, P <-- P(NE)		10101						*---
	W, P <-- P(E)		10110						*---
	NW, P <-- P(SE)		10111						*---

Table 3-1. BLITZEN PE Operation Codes

field is also used to set the length of the shift register. Operations coded in a single field cannot be performed concurrently. An instruction which sets the shift register length prevents the simultaneous assignment of values to P.

Code bits 8, 9, and A specify the operation to be performed by the adder, resulting in asssignment of sum and carry values to B and C. This field is also used to assign values to registers B and C.

Code bits B and C specify shift register operations. Code bits D and E assign values to register A. Code bits F and G specify the transfer of the data bus to register G, register K, or the sum-OR tree.

Code bits H and I specify memory and I/O operations. Memory transfers to or from the data bus are considered read or write operations. Transfers to or from the I/O bus are considered output and input operations.

3.2 The Operation Set

Table 3-1 lists all operation codes. A complete operation code word is formed by selecting codes from each of the fields. A value of all zeroes in a field represents a no-operation for that field. An entire instruction operation code of all zeroes is a no-operation.

A total of 58 operations are defined. Of these, 40 are maskable, 17 can be conditionally executed, and five allow local memory address modification.

Symbols are used in the last field of the operation code to represent either 0 or 1. An "*" is used for operations which may be masked. When an "*" is coded as a 1, the appropriate operations in that operation code word are considered masked by register G. The symbol "#" is used for operations which can be performed conditionally. When "#" is coded as a 1, the appropriate operations are conditional on register K . The symbol"@" is used for memory operations which can have addresses locally modified. When an "@" is coded as a 1, the memory operation will use local address modification.

Up to seven operations can be combined into a single instruction, representing seven concurrent activities. As an example of combining operations into one instruction, consider a step in an addition process where one operand is in memory, the other in the shift register, and the result is being placed in the shift register. Operations in one processing

The format shows several distinct fields specifying operations to be performed. This allows several operations to occur simultaneously in one processing cycle. When a register is both the source of a data bit and the target of an assignment which can change the register state, the original value of the register is used as the source value. State changes occur late in the processing cycle.

Most of the fields represent operations. The last field is special. It represents individual bits that identify local operations. Code bit L identifies masked P register operations, both logic and routing. When code bit L = 1, these operations are masked by the G register. When L = 0, these operations take place in all PE's. Code bit M identifies masked operations involving the adder, shift register, registers A, B, and C, memory write and input. When code bit M = 1, these operations are masked by the G register. When M = 0, these operations take place in all PE's.

Code bit N has two distinct purposes. In one use it identifies conditional operations on P and C. When code bit N = 1, these operations are conditioned by register K. When N = 0, these operations take place unconditionally. In the other use it distinguishes input operations that are column specific from those that broadcast a data item to all columns. When code bit N = 1 the data is broadcast, when N = 0 it is written into a specific column. This dual use of bit N precludes simultaneous interpretation as both a conditional and broadcast identifier. Priority is given to interpreting N for broadcast operations. Thus, conditional operations cannot be performed at the same time as input operations.

The final code bit, O, identifies memory operations with local address modification. When code bit O = 1, memory operations have the global address modified by certain bits from the shift register shift path. When bit O = 0, the global address is used without modification. For further discussion of these local control features see section 2.4.

Operation code bits 0, 1, and 2 identify the source of data for the data bus. The data bus source can be any of the single bit registers in a PE, or a bit from memory. When memory is the source, a single bit memory read operation is performed at the addressed location. A memory read should be coded with the value 00 in code bits H and I to prevent coding a conflicting memory write or I/O operation in the same instruction.

Code bits 3 through 7 specify the logic operation to be performed at P, or specify the routing operation to one of the eight nearest neighbors. This

3. Operation Codes

BLITZEN is an SIMD machine. All PE's receive the same instruction from a control unit. An instruction contains an operation code and address information for accessing memory and performing I/O transfers. Operations are defined in section 2 on the PE architecture. This section specifies operation codes for the machine.

3.1 Operation Code Format

Operations performed in the PE's in one processing cycle are governed by a single operation code. The 23 bit format of this code is shown in Figure 3-1.

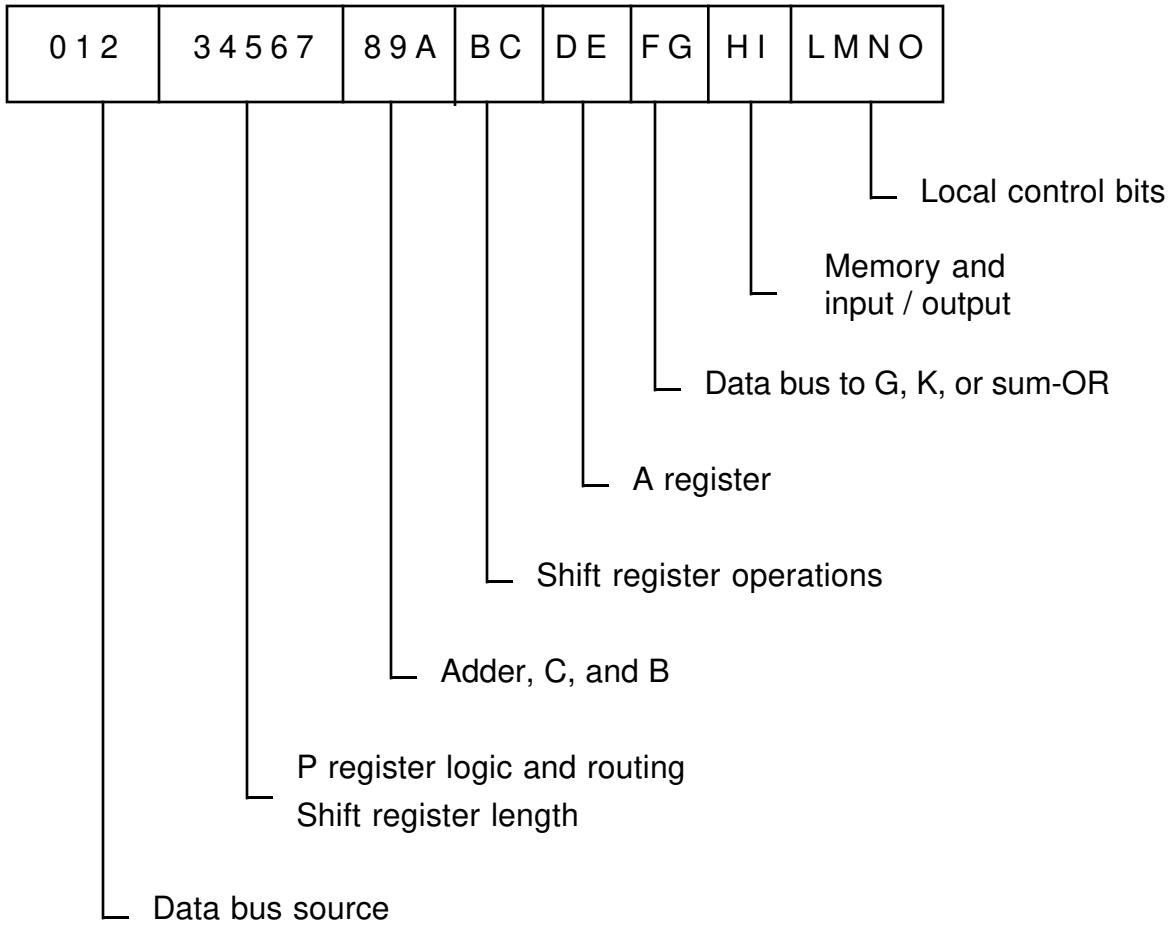

Figure 3-1. Operation code format

The shift path is the only multiple bit storage device other than memory and is therefore the only place a local address can originate. An address can be formed in the shift path. Recall that the 16 bit segment can be made inactive with respect to shift operations and the address information will remain in place. If a global address has all zeros, the OR will cause the local address to be the effective address. If a global address points to a data structure beginning on a power of two boundary, a value from the shift path can be used as a local offset into the structure.

2.5 Additional Comments on the Relation to the MPP PE

The BLITZEN processing element is based strongly on the MPP PE. Major changes were noted in section 1. Additional changes are mentioned here.

In BLITZEN, registers A and B have been given bidirectional connections to the data bus. This is necessary in conjunction with the change to a bidirectional shift register.

In the MPP, G is available to the data bus only via the "P equivalent to G" logic. The benefit of this signal is that a masked negate can be performed on data in memory in two cycles. In BLITZEN, the P equivalent to G logic has been removed but the same function can be performed via conditional logic. The function is set up by loading K with the mask rather than G (or by transfering G to K). In the first cycle the memory value is loaded into P using the conditional load feature that complements P where K = 1. In the second cycle write P into memory. Additional similar functions are available since all 16 logic operations at P can be done conditionally.

Several operations in the MPP are not included in BLITZEN. As discussed in the paragraph above, "P=G" is not available as a data bus source. The operation "clear parity error" is not included. In MPP, West to East S paths extend across chip boundaries to provide a way to shift I/O data from the edges of the array to internal locations. In BLITZEN, I/O busses provide the paths to internal elements so the long shift paths are not needed. Operations dealing with register S in the MPP do not appear in BLITZEN.

All of the processing functionality of the MPP has been retained. Additional features have been added in BLITZEN to improve performance.

2.4 Local Control Features

This section is intended to summarize the features provided in BLITZEN for individual PE's to control locally their activity. In a SIMD machine, identical control signals and addresses are distributed from a control unit to all PE's. In the absence of local control, all PE's would perform the same operation, accessing the same memory location in their respective RAM's. Performing the same operation in all PE's is the essence of SIMD parallel architectures. However, a certain amount of local control is almost essential and other local features can enhance performance.

The essential level of control is the ability to selectively turn off processing operations at individual PE's while allowing other PE's to perform. This is the masking feature implemented by a set of masked operations and a mask register, G. When a masked operation is issued, it will be performed in all PE's where G = 1 and not performed where G = 0. Register G can be set to indicate the result of a data dependent condition. Masked processing takes place in PE's where the condition is satisfied. This supports the high level IF - THEN conditional execution concept.

BLITZEN provides further local control through the use of a conditional operation test that complements certain operations where the condition is satisfied. In this case, different operations can be performed at the same time in different PE's. Register K is used to control conditional arithmetic and logic operations. When a conditional operation is issued, processing is normal in all PE's where K = 0. In those PE's where K = 1 the result of operations to load C or perform logic at P is complemented. As discussed in section 2.2.8, this feature improves performance in division algorithms. Since both normal and complemented operations take place, based on testing a condition, this is like the high level concept of IF - THEN - ELSE with both the THEN and ELSE clauses happening concurrently.

The final local control feature is the ability to modify a global memory address. Processors generally modify the address that appears in an instruction by adding an index or a base register value, or extracting an address from some location for indirect use. In a SIMD machine, logic that handles local modification of addresses must appear at each PE and be locally decoded. That is, the logic must appear at 128 PE's in this chip. To conserve chip area the modification chosen is the logical OR of certain shift register shift path bits with the global address.

increased bandwidth in BLITZEN resulting from this approach with I/O from each of the chips in a system. There is a total of 4096 bidirectional I/O paths. Even with this increase the ratio of processing bandwidth to I/O bandwidth is four-to-one, where conventional architectures have a ratio closer to one-to-one.

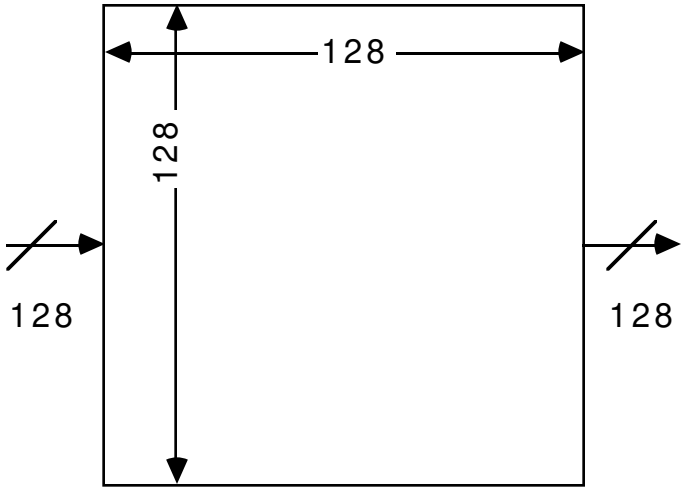

Figure 2-3. Edge oriented I/O in the MPP

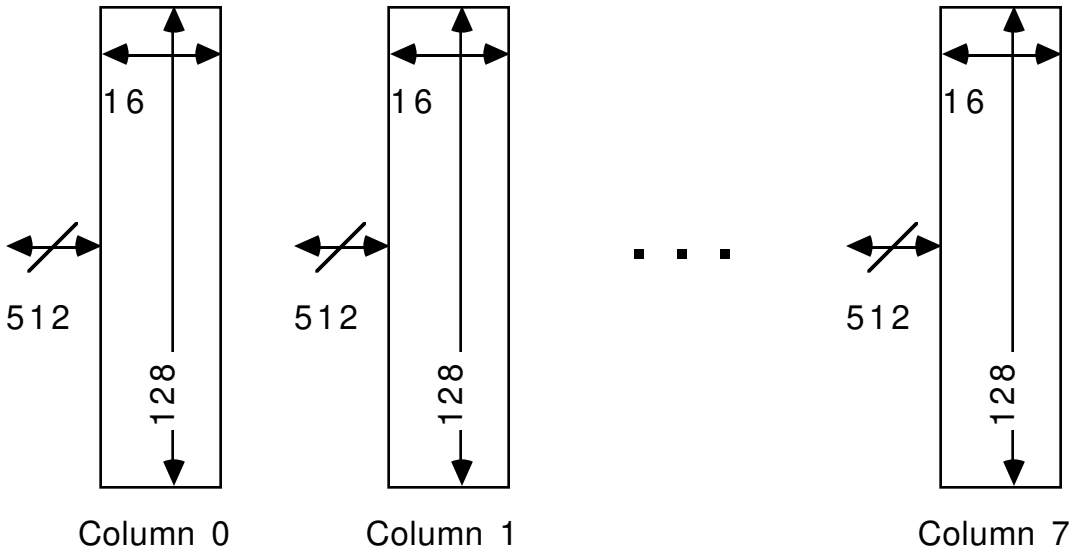

Figure 2-4. Interleaved I/O paths in BLITZEN

ten most significant bits of the 16 stage segment of the shift register. can be logically OR'ed with the global address, under control of locally modified memory operations. Such memory operations are selected by an operation code control bit. When normal (unmodified) memory operations are issued, the global address is unchanged.

The use of OR modification is a chip area saving approach, since an indexing feature would require a ten bit adder at each of 128 PE's on the chip. An OR can simulate indexing when data structures begin on power of two boundaries where the least significant bits are zeroes.

2.2.10 Sum-OR Tree

The value on the data bus can be sent to the sum-OR tree. An inclusive-OR of data bus values from all PE's on the chip is formed and brought to a chip pad for output. The sum-OR value is useful in associative processing algorithms.

Due to the propagation delay through the tree, this operation may take longer than one processing cycle.

2.3 Input / Output

High bandwidth I/O is provided by eight four-bit I/O buses per chip. Each bus serves16 PE's in a row on the chip with one PE selected for column specific input. A "broadcast" alternative allows the four bit value on the bus to be written into all PE's in a row. Figure 4-2 in section 4 shows the chip architecture with the I/O buses. In one processing cycle, 32 bits can be accessed at a column of eight PE's. The column is randomly addressable on each cycle.

Motivation for this scheme is based on the desire to bring the I/O rate into reasonable balance with the processing rate. VLSI layout aspect ratios influenced the design. Since it is impractical to layout a 1024 x 1 memory as a long, narrow area, the physical layout is more nearly square and will access four bits at a time. The bus oriented scheme takes advantage of the access width.

The anticipated system architecture will use RAM chips as the external devices connected to the processor chip I/O pins. This will be somewhat analogous to the staging memory in the MPP. Figure 2-3 shows MPP I/O paths numbering 128 inputs and 128 outputs. Figure 2-4 shows the

For a conditional operation, where K = 0, the result is normal. If K =1 the operation is modified to produce the complement of the normal result. Conditional operations are limited to setting or resetting C and to logical operations that load P.

Register K can be loaded from the data bus. It is one of the sources of data to the data bus. The operation to load K can be masked. A masked operation takes place only in the PE's where the G register is set.

2.2.9 Memory

A random access memory (RAM) is associated with each PE. From a processing point of view it is a 1024 x 1 bit RAM. A memory read operation reads the single bit specified by a ten bit address and places the value on the data bus. A memory write operation writes the value on the data bus into the location specified by a ten bit address. The memory write operation can be masked. A masked write takes place only in the PE's where the G register is set.

Input / output operations view the memory as a 256 x 4 bit RAM. I/O operations access memory as four bit items using the eight most significant bits of the ten bit RAM address. Four bit items are aligned with the least significant bit on the four-bit address boundary. Higher order bits are in higher addressed locations. An input operation transfers four bits from the I/O bus to memory at the addressed four-bit location. An output operation transfers four bits from memory, at the four-bit location, to the I/O bus. Input operations can be masked. A masked input takes place only in the PE's where the G register is set.

BLITZEN I/O operations are unusual for a SIMD machine in that they apply to a subset of all the PE's on a chip. A four bit column address is used to select, under program control, one of 16 columns of PE's on the chip. In addition to column specific I/O, there is a broadcast input operation that writes data from each four bit bus into all PE's on the row. The column select address is ignored for broadcasting. This feature is useful for distributing a constant value to all PE's for array operations with one scalar operand.

In a SIMD machine, the control unit issues an instruction to all PE's. If a memory operation is involved, one address is delivered to all PE's. Signals from the control unit are considered "global" to the system. BLITZEN has a feature which allows local modification of global memory addresses. The

either the data bus, the shift register, or the adder sum output. It can be cleared to zero. The value in B can be shifted right into the shift register. It is one of the sources of data to the data bus.

Operations which assign a new value to B can be masked. A masked operation takes place only in the PE's where the G register is set.

2.2.7 G Register

In massively parallel architectures operating as SIMD machines, it is useful to be able to enable or disable activity at individual PE's. This feature is commonly called "masking". In BLITZEN, the mask that enables certain PE operations is maintained in the G register. In those PE's where G is set (G = 1), masked operations are performed. Where G is reset (G = 0), masked operations do not take place. Since each PE has a G register, this represents a control feature local to each PE.

Operations on the P register are masked by setting an operation code bit. A second operation code bit is used to select masked operations on registers A, B, C, the shift register, and memory. This allows any combination of masked or unmasked operations from one group to be performed concurrently with masked or unmasked operations from the other group. There is, however, only one mask in register G.

The G register can be loaded from the data bus. If a load G operation is combined with a masked operation in the same processing cycle, masking is governed by the state of G at the beginning of the cycle. Register G is one of the sources of data to the data bus.

2.2.8 K Register

Local control of arithmetic and logic is supported through conditional operations that test K. The motivation for this feature is to allow some PE's to perform addition while others are performing subtraction. This is useful in implementing non-restoring division where the next step depends on the sign of the result of the current step. Sign information can be loaded into K. The approach is to form, conditionally, the two's complement of the second operand, thereby changing an addition algorithm into subtraction or a subtraction into addition. This requires control of the initial state of register C and the logical complement of values loaded into register P.

are not affected by length changes.

In one processing cycle the register shifts one place, either right or left. A right shift is from register B towards register A. In a right shift, the leftmost active stage is loaded with the value in B at the beginning of the cycle and the rightmost active stage is shifted out towards A. Register A can be loaded with the value shifted from the shift register, under program control. In a left shift, the rightmost active stage is loaded with the value in A at the beginning of the cycle and the leftmost active stage is shifted out towards B. Register B can be loaded with the value shifted from the shift register, under program control. The shift register can be cleared, however only the active stages are affected by the clear operation.

Masking applies to shift register operations. An unmasked shift operation takes place in all PE's. A masked operation takes place only in those PE's where the G register is set. For a masked operation, if G is reset the shift register is not disturbed. An operation code bit controls selection of unmasked or masked shift operations. The same bit controls operations that affect the adder and A, B, and C registers.

Figure 2-2 also shows a ten bit bundle labelled "local address modification bits". The bits are used to provide local modification of the global address supplied from a control unit, as discussed in sections 2.2.9 on the random access memory and 2.4 on local control features. The local address modification bits are the ten most significant bits of the 16 bit section of the shift register.

2.2.5 The A Register

This register supplies one of the inputs to the adder and is the rightmost stage of the shift register shift path. Register A can be loaded from the data bus, loaded from the shift register, or cleared to zero. The value in A can be shifted left into the shift register. It is one of the sources of data to the data bus.

Operations which assign a new value to A can be masked. A masked operation takes place only in the PE's where the G register is set.

2.2.6 B Register

This register receives the sum bit from adder operations and is the leftmost bit of the shift register shift path. Register B can be loaded from

unconditional or conditional, are subject to masking. Conditional operations cannot be combined with data input operations due to the use of a common operation code bit to specify both conditional and broadcast input operations.

2.2.4 Shift Register

The shift register is a variable length device with bidirectional shift paths. It is logically positioned between registers A and B in the path for recirculating a sum back to the adder input. The shift register has N stages where N can be set to 2, 6, 10, 14, 18, 22, 26, or 30. Registers A and B add two stages to the path length, providing a total length ranging from four to 32 in multiples of four.

Figure 2-2 is a diagram of the shift register and the registers on either end. The figure shows the structure of the shift register with groups of stages that can be bypassed to achieve variation in the shift path length. The path length is set by an operation which specifies a particular value for N. An operation to set the length becomes effective on the next processing cycle and holds for all subsequent cycles until another length setting operation is executed.

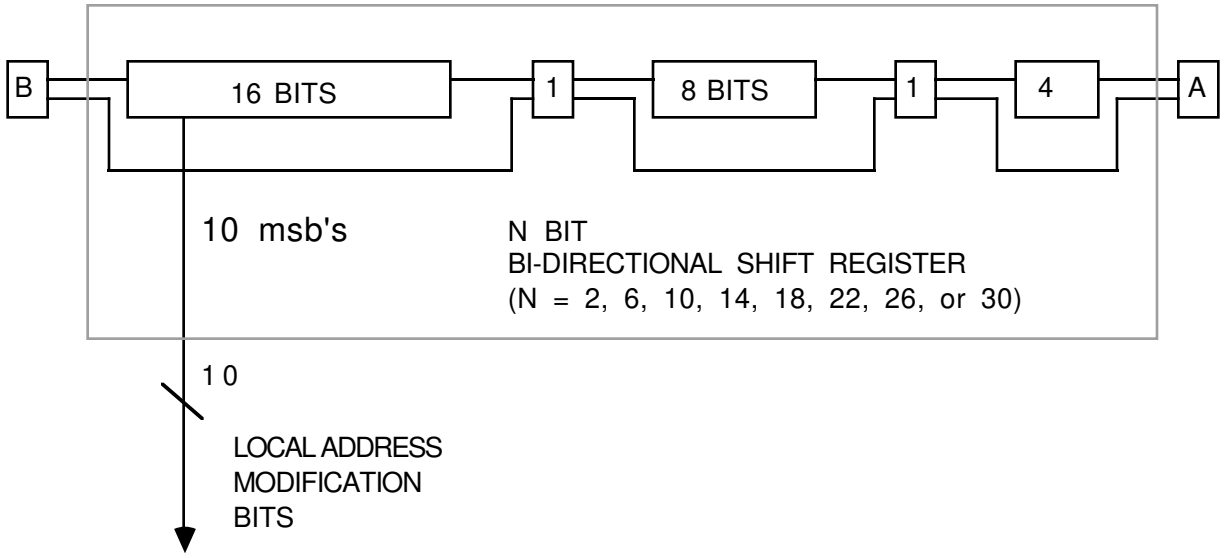

Figure 2-2. Shift register structure

Stages in the shift path for the length currently set are considered to be active stages. Stages that are bypassed are inactive. Shifting occurs only in the active stages. Data is maintained in position in the inactive stages. Due to the variable length structure of the register, the relative position of data bits may be rearranged if the length is changed. Values in A and B

as a sum bit, which is stored in the B register, and a carry bit, which is stored in the C register. Table 2-1 is the truth table for this operation.

A half add operation is also provided. This forms the two bit sum of the two bits stored in registers A and C. The state of P is not involved. The result is output from the adder circuit as a sum bit and carry bit, which are stored in B and C respectively. The truth table for a half add operation is given in Table 2-2.

Initial states			New states	
A	P	C	C	B
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Table 2-1. Full addition truth table

Initial states		New states	
A	C	C	B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 2-2. Half addition truth table

The full add and half add operations can be unmasked or masked, as selected by an operation code bit. An unmasked operation takes place in all PE's. A masked operation takes place only in those PE's where the G register is set. For a masked operation, if G is reset the states of B and C are not changed. A different operation code bit controls masking of P register operations. Thus unmasked or masked P register operations may be combined with unmasked or masked arithmetic operations in one processing cycle.

The state of the C register can be set (C = 1) or cleared (C = 0). These operations are also subject to masking.

Operations to set and reset C can be unconditional or conditional, as selected by an operation code bit. Unconditional arithmetic operations proceed as described above. For conditional operations, the condition tested is the state of the K register. If the K register is set, the operation is modified to produce the complement of the normal result in C. If K is reset, the result in C is normal. Arithmetic operations, whether

between the state of P at the beginning of a processing cycle and the value of the data bit on the data bus. The result of the operation is stored in P at the end of the cycle. All 16 Boolean functions of two variables are implemented. They are specified in section 3 on the operation set.

PE's are arranged in a two dimensional grid with neighbors in the eight compass directions N, NE, E, SE, S, SW, W, and NW. A routing operation transfers the state of P to the P register of a neighboring PE and accepts a new state from the PE in the opposite compass direction. Four bidirectional routing connections are brought out of each PE from the four logical corners: NE, SE, SW, and NW. The connections intersect between PE's as shown in figure 4-1 in section 4 on the chip architecture. A routing path is established by an operation which sends data out in one direction and accepts data in from one of the remaining directions. As an example, routing in the north direction can be achieved by sending P out to the NE and accepting P in from the SE. The data value on the SE input originated in the PE to the south.

Logic and routing operations at the P register can be unmasked or masked, as selected by an operation code bit. An unmasked operation takes place in all PE's. A masked operation takes place only in the PE's where the G register is set. For a masked operation, if G is reset the state of P is not changed. A different operation code bit controls masking of arithmetic operations. Thus unmasked or masked P register operations may be combined with unmasked or masked arithmetic operations in one processing cycle. Arithmetic operations do not affect the state of P.

Logic operations at the P register can be unconditional or conditional, as selected by an operation code bit. Unconditional logic operations proceed as described above. For conditional operations, the condition tested is the state of the K register. If the K register is set, the operation is modified to produce the complement of the normal result in P. If K is reset, the result in P is normal. Logic operations, whether unconditional or conditional, are subject to masking. Conditional operations cannot be combined with data input operations due to the use of a common operation code bit to specify both conditional and broadcast input operations.

2.2.3 Full Adder and C Register

The full adder performs bit serial arithmetic in the PE. A full add operation forms the two bit sum of the three bits stored in registers A, P, and C at the beginning of the processing cycle. The result is output from the adder

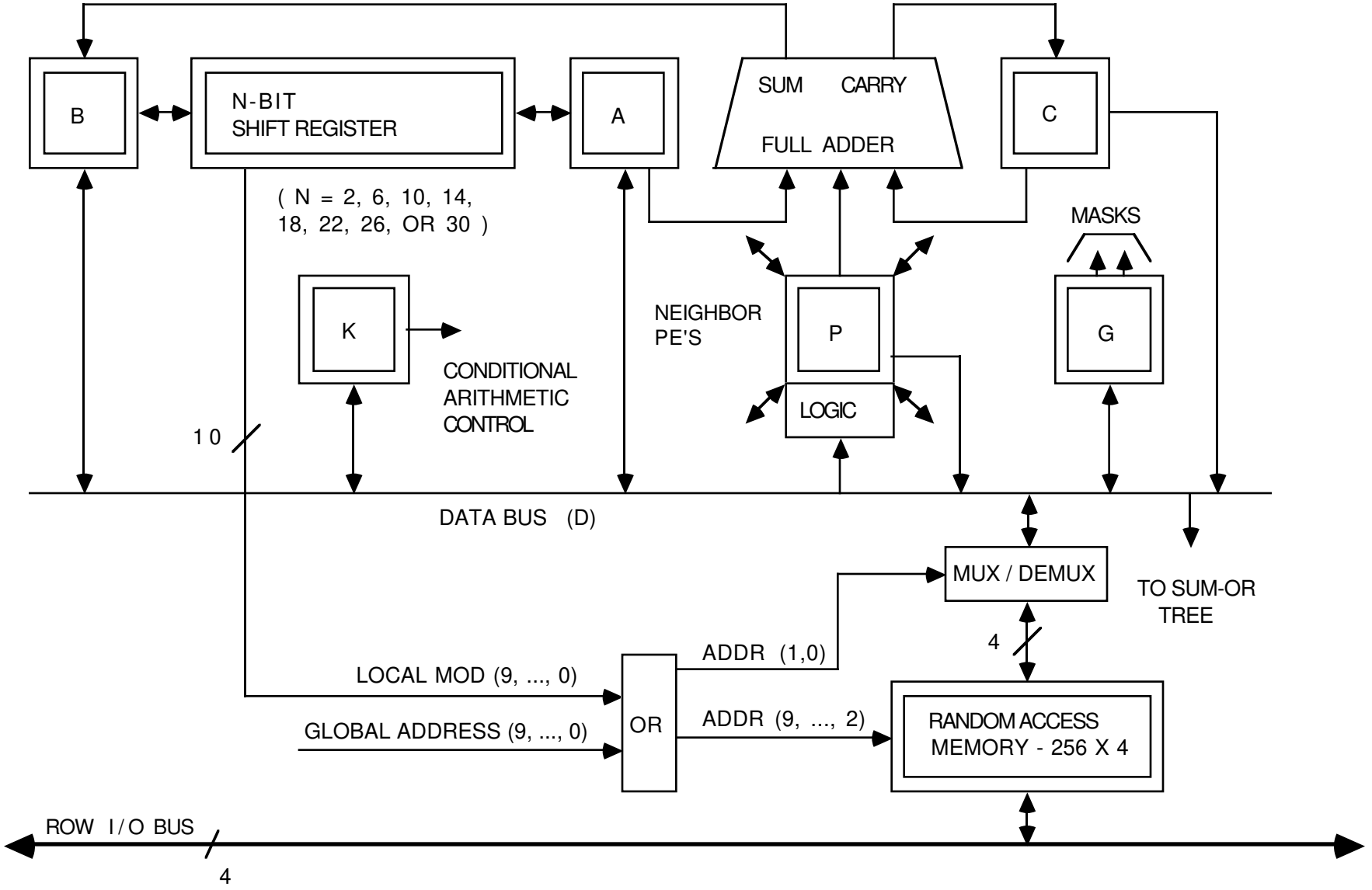

Figure 2-1. Functional units of one BLITZEN PE

2. BLITZEN Processing Element Architecture

This section describes one processing element (PE). The custom chip will have 128 PE's arranged in an 8 x 16 array, as discussed in section 4, Custom Chip Architecture. The system will have 128 chips for a total of 16,384 PE's arranged in a 128 x 128 array.

2.1 Overview

Figure 2-1 is a diagram of the functional units of one BLITZEN PE. Blocks with double line boundaries are storage devices. There are six single-bit registers labelled A, B, C, G, K, and P. Two devices hold multiple bits. One is a variable length shift register which, in conjunction with registers A and B, has a capacity of 32 bits. The remaining storage device is a 1024 bit random access memory (RAM). Arithmetic and logical operations are performed by a full adder and a logic block. The above elements communicate primarily over a single bit data bus. A four bit I/O bus provides a path to pads of the chip for connection to external storage devices. An I/O bus is shared among 16 PE's on a chip.

2.2 Detailed Description of Functional Elements

2.2.1 Data Bus

Elements of a PE communicate over the data bus. In one processing cycle it can transfer one bit of data from any one of its sources to one or more of its destinations. The source is: 1) one bit from the RAM, or 2) the state of the A, B, C, G, K, or P register.

Data on the bus may be transferred into: 1) the RAM, 2) the A, B, G, or K registers, 3) the logic associated with the P register, and 4) the input to the sum-OR tree. Several of these destinations may be specified in one instruction, as defined in section 3, Operation Codes.

The sum-OR tree forms the inclusive OR of all PE data busses on the chip. The single bit result is brought to a pad of the chip as an output signal.

2.2.2 P Register

The P register is central to arithmetic and logical operations in the PE. It is also used for inter-PE routing. For arithmetic, the state of P is one of the three inputs to the full adder. For logic, an operation is performed

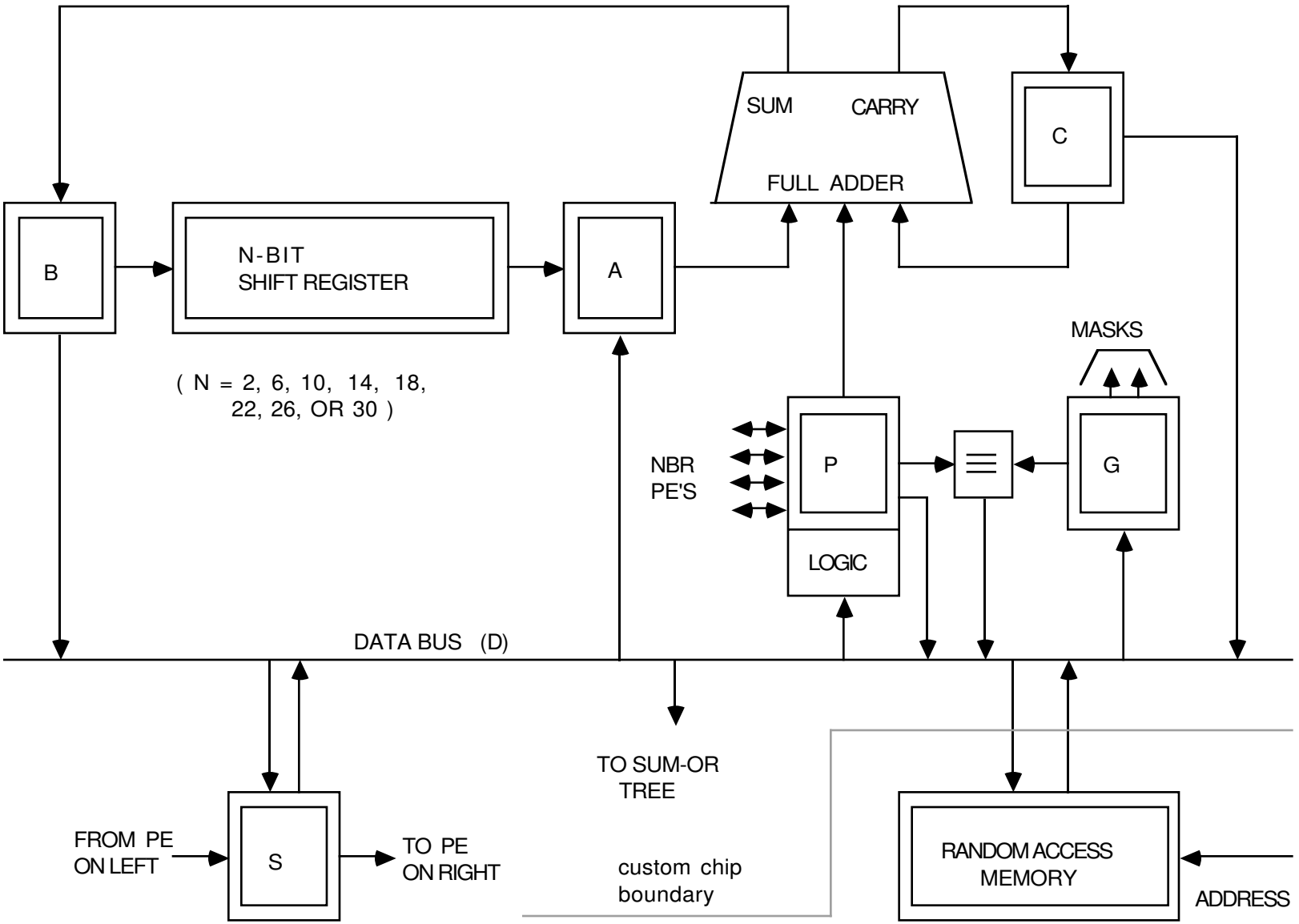

Figure 1-1. Functional units of one MPP PE

The effort reported in this paper is part of a project to design and fabricate a chip for BLITZEN, a new massively parallel computer system.

1.2 Improved Architecture and Technology

Experience with existing massively parallel machines has provided some insight to features that are useful and powerful. It has generated ideas on features that would increase the performance of such systems. Advances in integrated circuit technology have opened some possibilities in design and performance of processing element arrays. This project will incorporate architectural features and new technology toward the goal of a fast, dense chip for use in a high performance, physically small, massively parallel system.

1.3 Relation to the Goodyear Aerospace / NASA MPP

This project is using the MPP processing element architecture as a basic starting point. Figure 1-1 is a diagram of the functional units of one PE from the MPP. It includes a bit-serial processor and a random access memory. In the MPP, one custom design chip has eight processors. The memory is implemented with commercially available random access memory (RAM) chips. A full description of the MPP PE is provided in [Batcher] and [Burkley]. Since the BLITZEN PE has many similarities to this earlier device, some of the descriptive information here applies to the MPP. The present report has similarities to a report published by Goodyear Aerospace Corporation, reference [GAC]. Figure 1-1 is a slight modification of a figure in that report. Parts of this text, dealing with strong similarities to the MPP, are slight modifications of that report.

Significant architectural features of BLITZEN that differ from the MPP are:

1) Incorporation of RAM on-chip for each PE.
2) Bus oriented I/O with a four bit path for each set of 16 PE's.
3) Local modification of RAM addressing.
4) Local conditional control of arithmetic and logic.
5) A bidirectional, variable length shift register.
6) An eight neighbor grid.

Processing Element and Custom Chip Architecture for the BLITZEN Massively Parallel Processor

(Revision 1)

Donald W. Blevins, MCNC
Edward W. Davis, NCSU
John H. Reif, Duke Univ.

1. Introduction

This paper is an architectural specification for the processing element of a massively parallel processor. It further describes the incorporation of an array of these processing elements into a custom VLSI chip having approximately one million transistors. The report will serve as the basis for design of the custom chip.

1.1 Massive Parallelism

Speedup of computation is an important goal in computer architecture research. One architectural direction being exploited to achieve significant speedup is the use of many processors driven by a single control unit. This is the single instruction stream, multiple data stream (SIMD) concept. Massive parallelism refers to this concept when the factor of multiplicity is very large, perhaps exceeding 10,000 processing elements (PE's). Coupled with the use of a massive number of PE's is the simplicity of an individual PE. They operate in bit-serial fashion.

VLSI technology is essential to the realization of massively parallel systems. The technology has been used to produce several different processing element array chips that can be used as building blocks in massively parallel systems. Several systems have been built and are in use by industry, government agencies, and research organizations, including universities. Prominent systems are the Massively Parallel Processor (MPP) built by Goodyear Aerospace Corporation (now Loral Systems Group) for NASA Goddard Space Flight Center, the Connection Machine by Thinking Machines Corporation, and the Distributed Array Processor (DAP) by the British firm ICL.